Interoperable Access and Usage Control of Self-Sovereign Digital Twins Using ODRL and I4.0 Language

Jiahang Chen\textsuperscript{a,}, Lennart Schinke\textsuperscript{b,}, Xuebilian Gong\textsuperscript{c,}, Martin Hoppen\textsuperscript{d} and Jürgen Roßmann\textsuperscript{e}

Institute for Man-Machine Interaction, RWTH Aachen University, Ahornstr. 55, 52074 Aachen, Germany

Keywords: IoT, Interoperability, Access Control, Usage Control, Self-Sovereign Digital Twins, ODRL, I4.0 Language.

Abstract: The trend in digital transformation catalyzes an increasing amount of Digital Twins (DTs) being interconnected to share data and services. In this context, secured interconnections of DTs are a key foundation for establishing a trustworthy environment, which necessitates fundamental technologies and concepts regarding access control. Considering the layer of usage restrictions of data and services, traditional access control can be extended to usage control. Here, diverse policy models utilized to formalize access and usage control result in a lack of interoperability, especially in a decentralized Internet of Things (IoT). To address this issue, we propose in this paper a concept that applies Open Digital Rights Language (ODRL) to describe access and usage control policies in an interoperable way. Besides, we define a message-based communication protocol based on Industry 4.0 (I4.0) language to flexibly enable interoperable interactions with policy engines. The proposed concept is then integrated in an access and usage control management system and demonstrated in a proof-of-concept manner. Here it is also shown why the proposed concept forms a basis for the implementation of self-sovereign Digital Twins (SSDTs).

1 INTRODUCTION

In the era of digital transformation, the explosion of Digital Twins (DTs) (Singh et al., 2021; Tao et al., 2018; Liu et al., 2021) in scale reveals their pivotal potential in reshaping innovation across diverse domains (Jones et al., 2020). These virtual counterparts, mirroring physical entities, are increasingly interconnected to facilitate the seamless and transparent exchange of data and services, forming the backbone of technological advancement (VanDerHorn and Mahadevan, 2021). Based on the concept of self-sovereign identities (Preukschat and Reed, 2021), self-sovereign Digital Twins (SSDTs) (Chen et al., 2023) are proposed to promote a paradigm shift in control dynamics. SSDTs empower assets to exercise sovereignty, e.g., autonomy and authority in managing interconnections, using their digital counterparts. When being interconnected, communication participants can be generally referred to as subject (i.e., the initiator of the communication) and object (i.e., respondent of the communication). Ensuring security and interoperable access and usage control emerges as a cornerstone for establishing trust and reliability (Alam et al., 2011). Access control determines which parties can access protected resources (Samarati and de Vimercati, 2000). By integrating the layer of usage restrictions, access control can be extended to usage control that defines conditions (e.g., temporal and quantity limitations) regarding resources’ usage (Park and Sandhu, 2002). In both generalized DTs and our specific SSDTs, robust access and usage control mechanisms are indispensable for safeguarding communication in terms of confidentiality, integrity and availability of data and services. Our vision is interpreted in Figure 1. Here, SSDTs are positioned in the center to manage interconnections (e.g., from humans or other SSDTs) with the purpose of accessing the protected data resource. Additionally, it is also possible to access the policies associated with the resource. This is feasible because SSDTs can process this access due to their full autonomy and authority over their data and the attached interoperable policies.

Nonetheless, the state of the art faces a critical
challenge since several years - interoperability - with respect to policies and policy engines, see (Patil et al., 2007; Alam et al., 2011; Azeez and Venter, 2013). According to (Lee et al., 2021), there have not been any standards proposed to enhance policy interoperability. Although there are standardized models for expressing access and usage control policies, it is conceptually intricate for SSDTs governed by different policy models to authorize each other effectively. The reason for this is that they are not able to understand the structures and semantics of other access control models (ACMs) (Alam et al., 2011; Esposito, 2018). For instance, an SSDT equipped with Role-based Access Control (RBAC) (Sandhu, 1998) cannot be requested to enforce authorization in a landscape where SSDTs are managed using Attribute-based Access Control (ABAC) (Hu et al., 2015). In fact, there are also proposals which require a transformation of access models into a certain one, like (Hafeez et al., 2012). However, this increases the expense of implementation and enforces an understanding of the transformation process. The same issue can be found in the realm of usage control models (UCMs), see Section 2.2. In addition, interacting with policies varies depending on the implementation of the system architectures. Overall, the absence of interoperability may result in a fragmented Internet of Things (IoT) ecosystem where data or functionalities cannot be flexibly shared or reused, thereby limiting data analysis and value extraction.

The emergence of Open Digital Rights Language (ODRL) (Ianella, 2007), a standard language for expressing digital rights regarding permissions, obligations, and constraints, provides capabilities to enable a formal description of policies. To bring the communication participants to the same level of interoperability, a standardized interaction method is required, e.g., via a message-based communication protocol. The Industry 4.0 (I4.0) language standardized in VDI 2193-1 (Belyaev and Diedrich, 2019) delivers structures and vocabularies of messages (Kanaan et al., 2023). This language is primarily intended for interactions between I4.0 components. But due to its standardization, modular design, extendability, and the adaptability for efficient data exchange (Belyaev and Diedrich, 2019), the language can always be applied for constructing the structures and types of communication messages for other uses (BMWi, 2018).

In this paper, our contribution is summarized as follows. First, we present a conceptual framework to enable formalization of interoperable access and usage control policies on the basis of ODRL, allowing to express rich rules and integrate various vocabularies. For the interoperable communication of SSDTs regarding policy access, we define a message-based communication protocol leveraging I4.0 language. We then introduce an integrated access and usage control system architecture that incorporates the proposed ODRL-based framework. Finally, we show how the proposed concept provides a basis for SSDTs.

The presented paper is structured as follows: Section 2 summarizes the state of the art. Followed by Section 3, a concept is proposed. We demonstrate our proof of concept in Section 4 and conclude the paper in Section 5.

2 STATE OF THE ART

An overview of the state of the art is provided in this section. This is intended to establish a basis for understanding and to identify the problem addressed in the paper. Thus, Section 2.1 deals with ACMs and Section 2.2 with UCMs. This is followed by an introduction to Digital Rights Expression Languages (DRELs) in Section 2.3. Finally, this section closes with a discussion of the interaction with policies and policy engines and the associated problems in Section 2.4.

2.1 Access Control Models

ACMs are utilized to formally structure and describe policies that determine which parties are allowed to access protected resources. Different types of ACMs can be broadly categorized according to the right assignment mechanisms. In the following, we introduce some of the widely-used ACMs (Zhang, 2023):

- In Discretionary Access Control (DAC), resource owners assign permissions $P$ directly to users $U$ (i.e., the identity of users) via $P \rightarrow 2^U$.
- Role-based Access Control (RBAC) (Sandhu, 1998) streamlines the management of permissions
of entities and resources. Authorization predicates restrict the attributes of the entity and/or item, e.g., name of entity must be "Frank" or type of item must be ".pdf". Obligations define actions that must be executed by an entity before, during or after using an item, e.g., an entity must delete a file after reading it. Conditions include environmental restrictions that must apply before or during usage, e.g., an item is available for use in the EU only (Park and Sandhu, 2004; Lazouski et al., 2010; Schütte and Brost, 2018).

The use of UCON_ABC models facilitates the formal expression of diverse policies and requirements. Nevertheless, the practical implementation may proceed in various architectures or mechanisms (Lazouski et al., 2010), as the aspect of policy implementation is not covered in this framework. Interoperability between different implementations remains an analytical concern that necessitates further examination within the scope of deployment.

2.3 Digital Rights Expression Languages

DRELs (Barlas, 2006) are languages designed for expressing rights (e.g., permissions and constraints) associated with digital content. MPEG-21 REL (Wang et al., 2005), based on XML, enables the standardized representation of digital rights associated with multimedia content. The use of MPEG-21 REL is quite limited in access and usage control due to its specificity for multimedia domain. XrML (designed for digital media, e-commerce, and intellectual property management (Wang et al., 2002)) and RightsML (designed for news and publishing industry (Kasdorf, 2004; Lazouski et al., 2010; Schütte and Brost, 2018)) are faced with the same problem due to application restrictions. In XACML, there is an official policy language model used to express obligations and advice in addition to permissions. However, this model doesn’t inherently support several ACMs like RBAC (Ferrini and Bertino, 2009). Meanwhile, the use of XACML represents a security risk for those with a propagation feature to grant privileges to another user, e.g., DAC and CapBAC (Ferraiolo et al., 2016). In the context of UCMS, the XACML policy model necessitates enhancements to capture the UCON_ABC model adequately (Colombo et al., 2010). This involves extending the original XACML policy language model to integrate attribute mutability, persistent policy evaluation, and to set conditions over ongoing attribute updates and obligations. Overall, for the applicability of ACMs and UCMS in XACML, modifications are necessary to cater the authorization requirements especially in decentralized landscapes, as stated in (Masood et al., 2012).
Literature review reveals the possibilities of using the ODRL information model to express different ACMs. (Esteves et al., 2021) extend the ODRL profile to model ACL to specify and enforce individual's data sharing preferences. (Alishamisi et al., 2023) propose an approach to improve RBAC capability towards usage restrictions using the ODRL profile. However, based on our observation, most research primarily focuses on the use of ODRL vocabulary and ontology to enrich ACMs towards usage control (Steyskal and Polleres, 2014). Only fewer consider the adoption capabilities of ODRL for expressive ACMs. In the realm of UCMs, a variety of policy languages in the context of International Data Spaces (IDS) (IDSA, 2023) ecosystems is notable. MYDATA usage control technology has devised its own XML-based policy language to articulate data usage restrictions (FraunhoferIESE, 2023). Concurrently, there is also a policy language known as LUCON, designed for controlling data flows between endpoints (Eitel et al., 2021). The use of these technology-dependent policy languages by different organizations makes mutual understanding and cooperation among various UCMs challenging. In IDS, stakeholders have concurred on adopting ODRL, where policies are referred as Specification Level Policies, as opposed to technology-dependent policies (e.g., MYDATA, LUCON) as so called Implementation Level Policies (Hosseinzadeh et al., 2020; Jung and Dörr, 2022).

2.4 Interaction with Policies and Engines

Interactions with policies and engines rely on the deployed system architectures, generally via APIs or message-based protocols. In Open Policy Agent (OPA) (OPA, 2023), communication aligns with the defined OPA-specific REST APIs. Similarly, (Moghaddam et al., 2016) proposed a policy management engine model that is reached via APIs. This means, access to them requires an obligatory use and understanding of the specific APIs. In XACML, details of the communication protocol and message formats may vary between different XACML implementations. (Lee et al., 2015) proposed a RBAC system for substation automation systems using XACML. Here, the message-based communication protocol is defined highly compatible to IEC 62351, a standard for substation data security regarding RBAC. In LUCON, data flows are controlled through messages and a rule-based engine, with a primary focus on message labeling and routing (Schütte and Brost, 2018). This means that there is still an absence of a comprehensive message-based protocol to cover semantics.

In the context of ODRL, there have been several publications focusing on communication protocols. (Arnab and Hutchison, 2005) proposed an approach to enable bi-directional message-based communication, allowing the end users to add, remove, and replace rights from ODRL policies. In IDS, there are defined subclasses of the odrl:Policy which allow an assigner to offer a policy, i.e., IDS:ContractOffer. As response to the offer, the assignee can either agree with the offer (i.e., via IDS:ContractAgreement) or suggest a modification (i.e., via IDS::ContractRequest). These proposals are derived on top of the ODRL information model. For message-based communication, we additionally consider other information, such as the sender’s endpoint and message identification. Meanwhile, the format and content of responses are also relevant for communication with the same understanding. Using ODRL directly for establishing messages is problematic because ODRL is not designed for messaging.

3 CONCEPT

Drawing upon the literature review summarized in Section 2, we propose a modeling concept with ODRL and 14.0 as the central abstraction.

3.1 Policies

The ODRL information model is selected as the base of the modeling concept. In the current version (ODRL 2.2), the class of rule (odrl:Rule) can be specified into subclasses odrl:Permission, odrl:Prohibition, and odrl:Duty. As mentioned in Section 2.2, UCMs can be seen as an extension of ACMs. Due to the similar foundations, the problems of ACMs explained below apply analogously to UCMs.

ACMs are differentiated according to the assignment principles of permission. To map the assignment principles to the information model, we refine assignee by specifying odrl:PartyCollection. Actually, the current ODRL information model already provides odrl:Constraint with the primary aim to refine odrl:Action, odrl:Asset, and odrl:Party. However, using this constraint, the policy cannot be satisfied with hierarchical modeling (such as subroles of a role). Moreover, types of ACMs are represented in an implicit manner, as there is no class explicitly illustrating their types.

In light of this consideration, we propose an extended modeling of the information model, see Figure 2. The classes colored with blue
are the original part of the information model, whereas the red part depicts our proposed extension. First, odrl:PartyCollection, which is associated with odrl:Rule via odrl:assignee, is specified to express right assignments of various ACMs and UCMs. In addition, we also consider the policy as a resource (i.e., subclass of odrl:Asset), allowing policies themselves to be managed under additional policies. Thus, we propose the specification of odrl:Asset into odrl-ex:ResourceAsset and odrl-ex:PolicyAsset.

In addition to supporting hierarchical and explicit modeling, the proposed concept enables the flexible combination of various ACMs and UCMs into one policy. Listing 1 illustrates an exemplary access control policy, restricting who is allowed to retrieve measured values of a temperature sensor. This policy combines RBAC and ABAC to define the access rights, see Line 6-29. Line 30-40 define who is allowed to retrieve as well as update the policy.

Listing 1: An example access control policy combined with RBAC and ABAC.

```
{ "@type": "set",
  "uid": "http://example.com/policy:471",
  "permission": [
    { "target": {
      "@type": "ResourceAsset",
      "@value": "http://example.com/sensor/temp",
      "@type": "xsd:string",
      "rightOperand": {
        "@value": "home"
      }
    },
    "action": "read",
    "assigner": "http://example.com/user:123",
    "assignee": [
      {
        "@type": "Role",
        "@value": "smart_home_manager",
        "implies": ["smart_home_member"]
      },
      {
        "@type": "Attribute",
        "leftOperand": "org:location",
        "operator": "eq",
        "rightOperand": {
          "@type": "xsd:string",
          "@value": "home"
        }
      }
    ]
  ]
}
```

3.2 Message-Based Protocol

The fundamental element of the proposed message-based protocol is VDI 2193-1, which standardizes a basic message structure and vocabulary for a "Language for I4.0 components”. In this context, a message combines a frame with so-called interaction elements. First of all, the frame comprises the basic elements necessary for communication, especially, identification of communication participants (sender and receiver) as well as identification and reference of individual conversations and messages. The interaction elements comprise the necessary payload data. VDI 2193-1 specifies these messages as a general basis for automated interactions between communication partners. Specific types of interaction scenarios are specified in terms of so-called semantic interaction protocols. A concrete example is a "bidding protocol" specified in VDI 2193-2. Within the frame, VDI 2193-1 messages reference the applied semantic interaction protocol as well as the specific message type within the respective protocol.

Accordingly, we define four semantic interaction protocols to interact with policies. Here, every CRUD (Create, Update, Read, Delete) operation to access policies combines a request with a corresponding reply. Applying these protocols to VDI 2193-1 messages yields the message types as presented in Figure 3 within the namespace Interoperable Access and Usage Control (IAU).

The structured messages provide a standardized communication between different participants, ensuring each request will be responded with an appropriate reply. Moreover, the proposed concept places a strong focus on the request-reply pattern and status management via IAU:Status. The selection of this pattern contributes to the scalability and flexibility of the overall system, as new functionalities can easily be integrated. Status management triggered by replies allows for robust error handling and recording. Overall, the proposed concept contributes to creating a more organized, transparent, and reliable authorization system.
4 PROOF OF CONCEPT

Based on the concept outlined in Section 3, the subsequent parts delve into a practical demonstration to underscore the added value of our approach. Section 4.1 details the integration of our concept into an exemplary system architecture based on XACML. Furthermore, Section 4.2 focuses on the impact of our concept on facilitating trustful data sharing. To illustrate this, we present a data trustee as a use case, fostering an interoperable data sharing environment. Subsequently, Section 4.3 identifies parallels between the given use case and SSDTs. Here, it is shown why the proposed concept can serve as a basis for SSDTs and where their added value lies.

4.1 System Architecture

XACML offers a comprehensive system architecture that encompasses components like policy enforcement point (PEP), policy decision point (PDP), policy retrieval point (PRP), policy information point (PIP), etc. This reference architecture is widely used across different implementations in access control management systems. The proposed concept in Section 3 can be seamlessly integrated and used in this system architecture to enable an interoperable interaction regarding policies, see Figure 4. A gateway, representing a PEP, interrupts the data flow and forwards access requests as messages to a policy engine. The messages are created using the proposed message-based protocol, enabling creating, reading,
upgrading and deleting ODRL-based policies stored in the PRP. The policy engine then evaluates access requests by comparing the requests with the respective policies. In some cases, the evaluation must be supported by the PIP which provides relevant attributes associated with subjects. The ontology library provides, when needed, ontologies to semantically describe attributes used in PDP. The use of our concept enhances the richness of policy expression in this system. Moreover, the uniform understanding maintains a consistent and transparent access control environment across various interfaces and interactions within this system.

Figure 4: An exemplary XACML-based system architecture with the prototypical implementation of the concept provided in Section 3.

### 4.2 Impact on Data Trustees

Currently, various initiatives enable trustful data sharing while maintaining data sovereignty. These approaches often involve concepts such as data spaces and data trustees, mainly focusing on ensuring compliance with rights during data usage, based on UCMs (Schinke et al., 2023).

The concept of IDS addresses data sharing between two participants within a data space under a mutually agreed contract (Eitel et al., 2021). In practice, however, especially with machine-generated data, there are often several entities that are considered as data owners. This increased complexity explains the simplification in the use case in forestry, as stated in (Schinke et al., 2023). Here, there is only one provider that sends data to a data trustee to create offers on behalf of all owners. This process is shown in the upper part of Figure 5, where the red dashed lines represent the communication between the different owners (forest owner, contractor, and harvester) outside the data trustee’s purview. After executing the job, the harvester combines the generated data with individual usage rights to create various offers (different-colored squares with “O”). These offers are then sent to the data trustee. Based on this, actors can engage in negotiations to form a contract and thereby reach an agreement (different-colored hexagons with “A”), enabling them to access and utilize the corresponding parts of the data.

![Diagram](image)

Figure 5: Data trustee use case “trustful sharing of harvester production data” before (above) and after (below) the proposed extension of ODRL, use case based on (Schinke et al., 2023).

The specification of `odrl:Asset` into `odrl-ex:ResourceAsset` and `odrl-ex:PolicyAsset` proposed in Section 3.1 provides a solution to reduce complexity in the case of multiple owners. This makes it possible for an entity to independently append their usage rights to the provided data, by using the message-based protocol proposed in Section 3.2. The revised process is shown in the lower part of Figure 5, where the orange arrows and ellipses illustrate the new possibility. In this way, the harvester, responsible for data generation, is no longer burdened with tasks of combining the individual rights and
then creating offers. Instead, the harvester only transfers the data along with its own usage rights to the data trustee, initializing a policy through an IAU:CreatePolicyRequest. An excerpt of a possible corresponding message is shown in Listing 2. Subsequently, other stakeholders like forest owner and contractor can append their own usage rights to the policy using an IAU:SetPolicyRequest. An exemplary excerpt of a corresponding message is shown in Listing 3, where the attribute pathId defines the part of the policy to be set. Based on the defined policy, the data trustee then generates the resulting offers. Similarly, an IAU:SetPolicyRequest allows each owner to subsequently adjust their defined usage rights themselves, e.g., more or less restrictive.

Listing 2: Excerpt from an exemplary IAU:CreatePolicyRequest of a harvester.

```json
{
  "frame": { ... },
  "semanticProtocol": "https://example.com/IAU/CreatePolicy",
  "type": "request",
  "messageId": "harvester:123",
  "sender": {
    "identification": "123",
    "role": {
      "name": "harvester"
    }
  },
  "receiver": {
    "identification": "dataTrustee:456",
    "role": {
      "name": "dataTrustee"
    }
  },
  "replyBy": 123123123,
  "interactionElements": [ ],
  "policy": "$\{odrl policy acc. to Figure 2\}$
}
```

Listing 3: Excerpt from an exemplary IAU:SetPolicyRequest of a forest owner.

```json
{
  "frame": { ... },
  "semanticProtocol": "https://example.com/IAU/SetPolicy",
  "type": "request",
  "messageId": "harvester:123",
  "sender": {
    "identification": "123",
    "role": {
      "name": "harvester"
    }
  },
  "receiver": {
    "identification": "dataTrustee:456",
    "role": {
      "name": "dataTrustee"
    }
  },
  "replyBy": 123123123,
  "interactionElements": [ ],
  "policy": "$\{odrl policy acc. to Figure 2\}$
}
```

The proposed message-based protocol is also applicable for data retrieval. First, the consumer, e.g., a sawmill, needs to know the corresponding policy, which is achieved by sending an IAU:GetPolicyRequest to the data trustee. The trustee responds with an IAU:GetPolicyReply that contains the policy details or a rejection.

4.3 Portability to SSDTs

A closer look at Figure 5 shows that different types of participants can be involved in data sharing. In this example, people (forest owner, contractor), institutions (sawmill, research institute) and machines (harvester) are participating. As direct communication between the participants and the real harvester is not possible, this is handled by its DT. Currently, DTs enable data sharing based on a defined policy. However, to work as automated as possible, DTs should also be able to define and enforce policies. This is a basic idea of the SSDT. Using the proposed message-based protocol and integrating the system architecture presented in Section 4.1 in SSDTs forms a basis for this.

If all participants are represented as SSDTs, it is imaginable that, based on the proposed concept, they will be able to jointly define, negotiate and enforce policies in the future. This allows each SSDT to not only define who has the permission to use its data, but also to specify who has the permission to decide under which conditions its data can be used. For example, one SSDT could be responsible for financial negotiations on behalf of all owners, as it has a specialized negotiation model. A possible negotiation could be that the owners have agreed on a desired price of 100 € and a minimum price of 80 € for a dataset. A consumer now wants to use the data and offers 70 €. Through skillful negotiation, the SSDT agrees with the consumer on a price of 90 €. At the same time, each SSDT is able to share parts of the data that are sensitive only to itself on other terms without obtaining the permission of each owner. For example, an SSDT could share a corresponding part of the data with a research institute free of charge. Based on the ideas of an SSDT, this should be automated in the future. An active intervention by a human or a centralized component, such as a data trustee, is therefore no longer necessary.
5 CONCLUSION

The presented paper proposes a conceptual framework with ODRL and I4.0 language, making an advancement in managing access and usage control for SSDTs. The use and extension of ODRL achieves comprehensive and fine-grained structuring of policies, primarily enabling modeling of interoperable ACMs and UCMs. The granularity in representing policies not only refines the scalability, but also reduces the implementation complexity in practice, obviously enhancing the flexibility of authorization systems. The realization of the protocol rooted in VDI 2193-1 promotes the independence of communication participants in decentralized systems, indirectly decoupling the systems and increasing the fault tolerance of them. Standardized message formats enable systems built on different technologies to communicate seamlessly, basically enhancing the interoperability and flexibility in design of authorization systems.

To implement the concept in a proof-of-concept manner, the paper also demonstrates an application of a data trustee, highlighting the potential of our concept in facilitating interoperable and effective data sharing in complex digital ecosystems. In the subsequent phases of our research, we aim to establish an all-encompassing definition of the term SSDT. In addition to the core aspects that constitute an SSDT, specialized topics are also included. For example, we intend to develop a comprehensive approach to interoperably describe the relevant processes involved in authorization, specifically tailored as integral components of the authorization system of SSDTs. Furthermore, we plan to incorporate semantic web technologies into the framework to add semantic meanings to the system, making it more intuitive and effective in managing complex authorization scenarios.

REFERENCES


International Multitopic Conference (INMIC), pages 111–118. IEEE.


